UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/740,748	12/19/2003	Tin Qian	M1103.70168US00	4932

45840 7590 06/22/2010

WOLF GREENFIELD (Microsoft Corporation)
C/O WOLF, GREENFIELD & SACKS, P.C.
600 ATLANTIC AVENUE
BOSTON, MA 02210-2206

EXAMINER	
WANG, HARRIS C	

ART UNIT	PAPER NUMBER
2439	

MAIL DATE	DELIVERY MODE
06/22/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/740,748	QIAN ET AL.	
	Examiner	Art Unit	
	HARRIS C. WANG	2439	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *23 March 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *7 and 11-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *7, 11-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

Applicant argues "a local port is 'available' when an application is **instantiated** and **binds a socket to the local port.** Thus, instantiating at least one template is performed using the **stored local port**. For example, a....pg. 73 of Applicants' specification includes TransportTemplate, IPSecTemplate, and IPSECAuthorizationTemplate that are examples of the instantiation templates. None of the templates take a local port as a parameter. Instead, the templates use a respective "get" method to obtain the local port, which indicates that the local port condition is generated when an application is **initiated** and **binds a socket to the local port** (pg. 8 of Remarks)."

The claimed language requires "when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored...instantiating at least one template using at least the stored local port."

At the outset, the Examiner disagrees that the **omission** of a local port as a parameter in the **examples** of templates supports the Applicant's conclusion that the local port condition is generated when an application is initiated. In fact, the Examiner was unable to find the "get" method to obtain the local port at all.

Even assuming this was true there is nothing in the claim language describing the generation of a local port. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., generation of local port condition) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

Applicant next argues that "Terzis describes that, to associated rules with a received packet, information in the header of the packet can be used. As should be understood by one of skill in the art, this is different from storing a local port **when an application is initiated and binds a socket to the local port**...indeed, a packet is different from the application that binds a socket to the local port, as claimed (Remarks pg. 9)"

The Examiner respectfully disagrees. A packet can come into the system (seen in Figure 9) on behalf of an application. An HTTP incoming request is initiated from a web browser (which can be considered an application). HTTP would typically use port 80. Terzis gives multiple examples of Applications initiating and binding to ports throughout the specification especially Paragraph [0096] describes instantiating either IPSEC, SSL, PPTP and other secure tunneling templates.

The Applicant argues "The Office Action then contents that "this is similar to the creation of the at least one policy for the application based upon the N-tuples" However, in this portion, the reference again describes classifying a packet based on the existing

set of rules, rather than, when certain conditions are met, instantiating at least one template (Remarks pg. 9)."

The Applicant seems to argue that the terms "when certain conditions are met" is different from "based on the existing set of rules." The Applicant also appears to argue "classifying a packet (based on N-tuple elements, and afterwards processing based on associated rules)" is different from "instantiating the template ('The client instantiation templates are instantiated only when the full 5-tuple is available while the server instantiation templates only when the local 3-tuple is available" Applicants specification pg. 74). The Examiner respectfully disagrees.

The remaining arguments are derived from the above and are unpersuasive for the same rationale.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 7-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Terzis (20040243835) in view of Lambert (20020099952).

Regarding Claim 7,

Terzis teaches an object model for managing a service on a computer, the object model comprising:

A policy object model for specifying

by a first user, at least one first policy that the service supports in a packet-centric form *("the subsystems include a firewall…The firewall operates at layer 4 (transport)…The firewall serves to prevent unauthorized access of a network…by filtering out packets that originate from unauthorized users or sources. Performing filtering of packets can be effective in deterring certain types of unauthorized access attempts, but requires inspection*

of each packet" Paragraph [0089]) ("The resource access rules are used to control which users have access to what resources. The resource access rules define priority…The priority assigns a priority to the rule as each new incoming flow is evaluated against each of the policy rules according to their priority" Paragraph [0120]) and

by a second user, at least one second policy by selecting a security level from a plurality of security levels, with each security level from the plurality of security levels being previously set for a specified user *("the policy engine talks to the components on the data plane to install and remove filters in response to policy rules," Paragraph [0062]) ("The policies can be determined both by the identity of the user as well as by the group the user is associated with…Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services" Paragraph [0089]) ("The resource access rules are used to control which users have access to what resources. The resource access rules define…permission level" Paragraph [0120])* The Examiner interprets permission level as the security level.

"when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored, and, when parameters of the application match a condition in an application rule of the policy object model, at least one template is instantiated using at least the stored local port to create at least one policy for the application "

Figure 14 and associated text shows L7 (Application layer) rules DB. Also see Paragraphs [0086], [0089], [0117], [0129].

*(Figure 6 of Terzis shows the Policy Object class, **600**. Under the Policy Object is the Policy Component **610** and the Policy Rule **670**. One of the PolicyRules is ResourceAccessRule **675** which includes "AllowIdentifiers, DenyIdentifiers, and Log."*

*According to Paragraph [0105] Policy Object **600** is an "abstract base class." Paragraph [0118] teaches Policy Rules **670** is "an abstract class that all policy rules derive from."*

As such, Terzis teaches "wherein the policy object model comprises a plurality of policy action classes representing at least a deny, permit and log actions on the service of on at least one packet.")

(Terzis Paragraphs [0083-0084] and explanation in Response to Arguments)

A policy engine platform for interacting of the first user with the at least one first policy and of the second user with the at least one second policy, and to provide the at least one first policy and the at least one second policy to at least one component that performs the service.

("The policy interpreter interfaces to the SNMP Agent," Paragraph [0064], Fig 7.)

The Examiner interprets the policy object model as the "policy engine" and policy engine platform as "policy interpreter."

As seen in Fig. 7, the Policy Interpreter acts as an intermediary between the SNMP agent and the Policy engine. Because the purpose of a SNMP agent is to facilitate information between network components and the purpose of the policy engine is to provide policies, it is inherent that the policy interpreter will provide one or more policies of which one will actually perform the service.

Terzis teaches the policy engine platform comprises a rule editor that is configured by the first user to perform at least one of deleting, adding, editing the at

least first policy by the first user. *("The interface between the policy engine and the SNMP agent may be used to add and delete policy objects" Paragraph [0064])*

Terzis teaches a setting editor that is configured by the first user to select a security level from the plurality of security levels for the second user. *("an operator may be able to enter a set of human readable access rules that define what resources and services are accessible to that user (or machine). According to one embodiment, these human readable access rules are stored as policy objects." Paragraph [0136]) ("the policy engine talks to the components on the data plane to install and remove filters in response to policy rules," Paragraph [0062]) ("The policies can be determined both by the identity of the user as well as by the group the user is associated with…Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services" Paragraph [0089]) ("The resource access rules are used to control which users have access to what resources. The resource access rules define…permission level" Paragraph [0120])* The Examiner interprets permission level as the security level.

Terzis does not explicitly teach if it has been determined that the first user is authorized to perform the specification by comparing a rank of the first user against a permitted rank. The Examiner interprets a permitted rank as the priority level, as described by the Applicant in pg. 8 of Remarks, "A policy provider is associated with a particular priority class or level" (Paragraph [0051] of Specification).

Lambert teaches determining whether a first user is authorized to perform the specification by comparing a rank of the first user to a permitted rank before specifying

a policy. *("the group policy objects…may be provided by administrators per site, domain, organizational unit, group and user. Among other things, group policy technology also provides a flexible and hierarchical way in which each administrator can establish which policies will win out over others if multiple policies conflict. For example, site policies can be set up to prevail over domain policies, which in turn can be set up to prevail over organizational unit policies…." Paragraph [0080])*

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the object model of Terzis with the policy provider priority ranking system of Lambert.

The motivation is that Lambert teaches a well known way to deal with conflicts with group policy objects.

As explained by the Applicant in the interview, the application rule to be matched did not determine the policy only determined which template to use. The template would then create the policy.

This is supported by the by pages 73-74 of Exhibit C in the Applicant's specification. Also according to the interview, the Applicant said that support for "when an application is initiated and binds a socket to a local port… (Claim 1) " was taught by "On the other hand, if it is not a client instantiation, only local 3-tuple, i.e. local address, protocol, and local port, available" wherein the local port is available.

Terzis in paragraph [0083-0084] teaches observing an incoming packet for low identification data (e.g. **source port**, source IP address, **destination port**, destination IP address, IP protocol, VLAN-ID) within the header of the packet).

Terzis in the same paragraphs further teaches "Classification involves searching the N-tuples against a rule set." This is similar to the determining whether the application is 3-tuple or 5-tuple as described in pages 73-74 of the Appendix.

Terzis then teaches "After a frame has been classified its N-tuples and classicfication result are added to an identification database (an association is made). THe packet then proceeds to be processed based on the associated rules." This is similar to the creation of the at least one policy for the application based upon the N-tuples.

Therefore the Examiner believes that Terzis teaches "when parameters of the application match a condition in an application rule of the policy object model, instantiating at least one template using the at least stored local port to creat at least one policy for the application"

As such, Examiner believes these paragraphs teach the limitation "when an application....to create at least policy for the application" as clarified by the Applicant.

Regarding Claims 11 and 12,

Terzis and Lambert teach the object model of claim 7, Terzis further teaches wherein the policy engine platform comprises a setting editor configured to automatically generate a policy based upon an application and user combination,

wherein the setting editor generates a plurality of policies, and is further configured to permit said second user to select from the plurality of policies.

("After a user has successfully logged [in]...the Launch-pad module may contact the policy engine to receive the list of resources that are available to that user...Once fount the policy user may return each of the resources in those rules back to the Launch-pad module, Paragraph [0065])

Where the Launch-pad is defined as a user interface in Paragraph 100. The launch pad screen is capable of displaying "applications...that are specifically made available to that user (Paragraph 106).

The Examiner interprets the second user to be an administrator that implements user-centric policies. *(The resource access rules are used to control which users have access to what resources. Paragraph [0120])*

Regarding Claim 13,

Terzis and Lambert teach the object model of claim 12, Terzis further teaches wherein the setting editor is further configured by said second user to permit setting one of the plurality of policies as a default policy.

("generating, based on the access policies, at least one access rule for each of a plurality of security system sublayers," Claim 1)

The Examiner interprets the at least one access rule as the default policy.

The Examiner interprets the second user to be an administrator that implements user-centric policies. *(The resource access rules are used to control which users have access to what resources. Paragraph [0120])*

Regarding Claim 14,

Terzis and Lambert teach the object model of claim 7, Terzis further teaches wherein the policy engine platform comprises a rule explorer for providing a view of the at least one first policy and the at least one second policy.

Because the policy interpreter interfaces between the SNMP agent and the policy engine (Fig. 7) it is inherent that there will be a component that allows a view of one or more of the policies.

Regarding Claim 15,

Terzis and Lambert teach the object model of claim 7, Terzis further teaches wherein the policy object model comprises a policyrule object usable to generate policy, the policyrule object comprising a condition property and an action property, wherein a policy generated by the policyrule object is configured to perform an action in the action property responsive to a condition in the condition property being met. (Fig. 6, 670)

Regarding Claim 16,

Terzis and Lambert teach the object model of claim 7, Terzis further teaches wherein the service is a firewall service. *("According to one embodiment the rules are generated and installed at the firewall level" Paragraph [0019])*

Regarding Claim 17,

Terzis and Lambert teach the object model of claim 7, Terzis further teaches wherein the policy engine platform is configured to deny providing said one or more policies to the component if a requester is not authorized. ("*Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services." Paragraph [0088])*

Regarding Claim 18,

Terzis and Lambert teach the object model of claim 17, Terzis further teaches wherein determining whether a requester is authorized comprises comparing a provider rank for the requester against a permitted rank, and if the provider rank for the requestor does not meet or exceed the permitted rank, denying the requester. (Fig 6. 675, PermissionLevel)

The Examiner interprets the parameter PermissionLevel under the Resource Access Rules as rank. Where the PermissionLevel is checked against a permitted PermissionLevel and if the PermissionLevel does not meet or exceed the permitted rank, to deny the requestor.

Regarding Claim 19,

Terzis and Lambert teach a method of managing a service on a computer, the method comprising:

specifying, via a policy object model, by a first user , one or more policies that the service supports in a packet-centric form *("the subsystems include a firewall…The firewall operates at layer 4 (transport)…The firewall serves to prevent unauthorized access of a network…by filtering out packets that originate from unauthorized users or sources. Performing filtering of packets can be effective in deterring certain types of unauthorized access attempts, but requires inspection of each packet" Paragraph [0089])*, and, by a second user, at least one second policy by selecting a security level from a plurality of security levels, with each security level from the plurality of security levels being previously set for a specified application and a specified user; *("The policy engine talks to the components on the data plane to install and remove filters in response to policy rules," Paragraph [0062]) ("The resource access rules are used to control which users have access to what resources. The resource access rules define…permission level" Paragraph [0120])* The Examiner interprets permission level as the security level.

when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored, and, when parameters of the application match a condition in an application rule of the policy object model, at least one template is instantiated using at least the stored local port to create at least one policy for the application "

Figure 14 and associated text shows L7 (Application layer) rules DB. Also see Paragraphs [0086], [0089], [0117], [0129].

*(Figure 6 of Terzis shows the Policy Object class, **600**. Under the Policy Object is the Policy Component **610** and the Policy Rule **670**. One of the PolicyRules is ResourceAccessRule **675** which includes "AllowIdentifiers, DenyIdentifiers, and Log."*

*According to Paragraph [0105] Policy Object **600** is an "abstract base class." Paragraph [0118] teaches Policy Rules **670** is "an abstract class that all policy rules derive from."*

As such, Terzis teaches "wherein the policy object model comprises a plurality of policy action classes representing at least a deny, permit and log actions on the service of on at least one packet.")

(Terzis Paragraphs [0083-0084] and explanation in Response to Arguments)

and interacting, via a policy engine platform, of said first user at least one first policy specified in said packet-centric form, and of said second user with said one or more policies specified in said user-centric form and/or said application-centric form;

("the Launch-pad module may contact the policy engine to receive the list of resources that are available" Paragraph [0065]) ("The resource access rules are used to control which users have

access to what resources. The resource access rules define…permission level" Paragraph [0120]) The Examiner interprets permission level as the security level.

and providing, via the policy engine platform, said one or more policies to said at least one component that actually performs the service. *("Once found the policy engine may return each of the resources in those rules back to the Launch-pad module" Paragraph [0065])*

Terzis teaches "the subsystems include a firewall…The firewall operates at layer 4 (transport)…The firewall serves to prevent unauthorized access of a network…by filtering out packets that originate from unauthorized users or sources. Performing filtering of packets can be effective in deterring certain types of unauthorized access attempts, but requires inspection of each packet. (Paragraph [0089])." Terzis further teaches ""The policies can be determined both by the identity of the user as well as by the group the user is associated with…Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services" Paragraph [0089])

The Examiner interprets the first user to be an administrator that implements packet-centric policies. (*The security rules 690 may describe how packets matching the source, destination objects should be secured. Paragraph [0130])*

The Examiner interprets the second user to be an administrator that implements user-centric policies. *(The resource access rules are used to control which users have access to what resources. Paragraph [0120])*

Terzis teaches the policy engine platform comprises a rule editor that is configured by the first user to perform at least one of deleting, adding, editing the at

least first policy by the first user. *("The interface between the policy engine and the SNMP agent may be used to add and delete policy objects" Paragraph [0064])*

Terzis teaches a setting editor that is configured by the first user to select a security level from the plurality of security levels for the second user. *("an operator may be able to enter a set of human readable access rules that define what resources and services are accessible to that user (or machine). According to one embodiment, these human readable access rules are stored as policy objects." Paragraph [0136]) ("the policy engine talks to the components on the data plane to install and remove filters in response to policy rules," Paragraph [0062]) ("The policies can be determined both by the identity of the user as well as by the group the user is associated with…Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services" Paragraph [0089]) ("The resource access rules are used to control which users have access to what resources. The resource access rules define…permission level" Paragraph [0120])* The Examiner interprets permission level as the security level.

Terzis does not explicitly teach if it has been determined that the first user is authorized to perform the specification by comparing a rank of the first user against a permitted rank. The Examiner interprets a permitted rank as the priority level, as described by the Applicant in pg. 8 of Remarks, "A policy provider is associated with a particular priority class or level" (Paragraph [0051] of Specification).

Lambert teaches determining whether a first user is authorized to perform the specification by comparing a rank of the first user to a permitted rank before specifying a policy. *("the group policy objects…may be provided by administrators per site, domain,*

organizational unit, group and user. Among other things, group policy technology also provides a flexible and hierarchical way in which each administrator can establish which policies will win out over others if multiple policies conflict. For example, site policies can be set up to prevail over domain policies, which in turn can be set up to prevail over organizational unit policies…." Paragraph [0080])

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the object model of Terzis with the policy provider priority ranking system of Lambert.

The motivation is that Lambert teaches a well known way to deal with conflicts with group policy objects.

Regarding Claim 20,

Terzis and Lambert teach the method of claim 19, Terzis further teaches further comprising automatically generating a policy based upon an application and user combination. *"After a user has successfully logged into the MACSS, the Launch-pad module may contact the policy engine to receive the list of resources that are available to that user," Paragraph [0065])*

Regarding Claim 21,

Terzis and Lambert teach the method of claim 20, Terzis further teaches further comprising generates a plurality of policies, and permitting a user to select from the plurality of policies. *("Once found the policy engine may return each of the resources in those rules back to the Launch-pad module" Paragraph [0065])*

As described before the Launch-pad module is a user interface. Examples can be found in Fig. 4 and Fig. 5.

Regarding Claim 22,

Terzis and Lambert teach the method of claim 21, Terzis further teaches further comprising setting one of the plurality of policies as a default policy. *("generating, based on the access policies, at least one access rule for each of a plurality of security system sublayers," Claim 1)*

The Examiner interprets the at least one access rule as the default policy.

Regarding Claim 23,

Terzis and Lambert teach the method of claim 22, Terzis further teaches further comprising authorizing a user prior to allowing the user to select the at least one policy from the plurality of policies.

It is inherent that the system administrator is authorized prior to selecting one policy from a plurality of policies. *("A system administrator uses user interfaces...to create*

access/security rules that allow users access to specific network resources based on a variety of parameters" Paragraph [0056])

Regarding Claim 24,

Terzis and Lambert teach an object model embodied on a computer-readable medium for managing a firewall service on a computer, the object model comprising a policy object model used to specify, by a first user, one or more policies that the firewall service supports in a packet-centric form, and, by a second user at least one second policy by selectin g a security level from a plurality of security levels, with each security level from the plurality of security levels being previously set for a specified application and a specified user *("The resource access rules are used to control which users have access to what resources. The resource access rules define…permission level" Paragraph [0120],* The Examiner interprets permission level as the security level), the policy model comprising a policyrule object usable to generate policy *(Fig. 6, PolicyRule, 670)*, the policyrule object comprising a condition property and an action property, wherein a policy generated by the policyrule object is configured to perform an action in the action property responsive to a condition in the condition property being met.

It is inherent that the policy rule is configured to perform an action responsive to a condition being met.

Terzis teaches "the subsystems include a firewall…The firewall operates at layer 4 (transport)…The firewall serves to prevent unauthorized access of a network…by filtering out packets that originate from unauthorized users or sources. Performing filtering of packets can

be effective in deterring certain types of unauthorized access attempts, but requires inspection of each packet. (Paragraph [0089]).” Terzis further teaches ““The policies can be determined both by the identity of the user as well as by the group the user is associated with…Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services” Paragraph [0089])

“when an application is initiated and binds a socket to a local port, at least the local port from the socket is stored, and, when parameters of the application match a condition in an application rule of the policy object model, at least one template is instantiated using at least the stored local port to create at least one policy for the application “

Figure 14 and associated text shows L7 (Application layer) rules DB. Also see Paragraphs [0086], [0089], [0117], [0129].

*(Figure 6 of Terzis shows the Policy Object class, **600**. Under the Policy Object is the Policy Component **610** and the Policy Rule **670**. One of the PolicyRules is ResourceAccessRule **675** which includes “AllowIdentifiers, DenyIdentifiers, and Log.”*

*According to Paragraph [0105] Policy Object **600** is an “abstract base class.” Paragraph [0118] teaches Policy Rules **670** is “an abstract class that all policy rules derive from.”*

As such, Terzis teaches “wherein the policy object model comprises a plurality of policy action classes representing at least a deny, permit and log actions on the service of on at least one packet.”)

(Terzis Paragraphs [0083-0084] and explanation in Response to Arguments)

The Examiner interprets the first user to be an administrator that implements packet-centric policies. (*The security rules 690 may describe how packets matching the source, destination objects should be secured. Paragraph [0130]*)

The Examiner interprets the second user to be an administrator that implements user-centric policies. *(The resource access rules are used to control which users have access to what resources. Paragraph [0120]*)

Terzis teaches the policy engine platform comprises a rule editor that is configured by the first user to perform at least one of deleting, adding, editing the at least first policy by the first user. *("The interface between the policy engine and the SNMP agent may be used to add and delete policy objects" Paragraph [0064]*)

Terzis teaches a setting editor that is configured by the first user to select a security level from the plurality of security levels for the second user. *("an operator may be able to enter a set of human readable access rules that define what resources and services are accessible to that user (or machine). According to one embodiment, these human readable access rules are stored as policy objects." Paragraph [0136]) ("the policy engine talks to the components on the data plane to install and remove filters in response to policy rules," Paragraph [0062]) ("The policies can be determined both by the identity of the user as well as by the group the user is associated with…Based on the policies associated with that user, a set of specific access rules are generated that enable the subsystems to provide filtering and deny access to prohibited resources and services" Paragraph [0089]) ("The resource access rules are used to control which users have access to what resources. The resource access rules define…permission level" Paragraph [0120])* The Examiner interprets permission level as the security level.

Terzis does not explicitly teach if it has been determined that the first user is authorized to perform the specification by comparing a rank of the first user against a permitted rank. The Examiner interprets a permitted rank as the priority level, as described by the Applicant in pg. 8 of Remarks, "A policy provider is associated with a particular priority class or level" (Paragraph [0051] of Specification).

Lambert teaches determining whether a first user is authorized to perform the specification by comparing a rank of the first user to a permitted rank before specifying a policy. *("the group policy objects…may be provided by administrators per site, domain, organizational unit, group and user. Among other things, group policy technology also provides a flexible and hierarchical way in which each administrator can establish which policies will win out over others if multiple policies conflict. For example, site policies can be set up to prevail over domain policies, which in turn can be set up to prevail over organizational unit policies…." Paragraph [0080])*

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the object model of Terzis with the policy provider priority ranking system of Lambert.

The motivation is that Lambert teaches a well known way to deal with conflicts with group policy objects.

Regarding Claim 25,

Terzis and Lambert teach the object model of claim 24, Terzis further teaches further comprising an IPSecRule derived from the policyrule object, the IPSecRule

being configured to trigger an IPSec callout when an IPSec condition is matched, and to indicate configuration parameters for securing traffic related to the callout. (Fig. 14, 1440).

The services dispatcher connects to the launch-pad which connects to the policy engine.

Regarding Claim 26,

Terzis and Lambert teach the object model of claim 25, Terzis further teaches wherein the IPSecRule evaluates a standard 5-tuple to determine if a condition has been met. (Fig. 11)

Regarding Claim 27,

Terzis and Lambert teach the object model of claim 24, Terzis further teaches further comprising a KeyingModuleRule derived from the policyrule object, the KeyingModuleRule being configured to select which key negotiation module to use when there is no existing secure channel to a remote peer.

("The key exchange field specifies how keys are exchanged and determines what key parameters will be used." Paragraph [0130])

The Examiner interprets key negotiation as key exchange. The Examiner notes that the key exchange field is part of the security rules, which is part of the policy rules.

Regarding Claim 28,

Terzis and Lambert teach the object model of claim 27, Terzis further teaches wherein the KeyingModuleRule evaluates a standard 5-tuple to determine if a condition has been met. (Fig. 11)

Regarding Claim 29,

Terzis and Lambert teach the object model of claim 24, Terzis further teaches further comprising a IKERule derived from the policyrule object and configured to specify the parameters for carrying out Internet Key Exchange key negotiation protocol. (Fig. 14, IKE)

Regarding Claim 30,

Terzis and Lambert teach the object model of claim 29, Terzis further teaches wherein the IKERule evaluates a local address and a remote address to determine if a condition has been met. This step is inherent in IKE protocol.

Regarding Claim 31,

Terzis and Lambert teach the object model of claim 29, Terzis further teaches wherein the IKERule comprises an IKEAction action property that defines the authentication methods for performing Internet Key Exchange key negotiation protocol. *("The key exchange field specifies how keys are exchanged and determines what key parameters will be used." Paragraph [0130])*

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to HARRIS C. WANG whose telephone number is (571)270-1462. The examiner can normally be reached on M-F 9-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, EDAN ORGAD can be reached on (571) 272-7884. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Harris C Wang/
Examiner, Art Unit 2439

/Edan Orgad/

Supervisory Patent Examiner, Art Unit 2439